

MỤC LỤC

GIỚI THIỆU	1
CƠ SỞ LÝ THUYẾT.....	2
BỘ VI XỬ LÝ.....	7
XÂY DỰNG GIẢI THUẬT	11
THIẾT KẾ GIAO DIỆN ĐIỀU KHIỂN DÙNG C# FORM.....	15
KẾT QUẢ VÀ NHẬN XÉT CHẤT LƯỢNG ĐIỀU KHIỂN ĐỘNG CƠ HỆ THỐNG	16
NHẬN XÉT CHẤT LƯỢNG HỆ THỐNG	18
HẠN CHẾ VÀ HƯỚNG CẢI TIẾN.....	18

GIỚI THIỆU

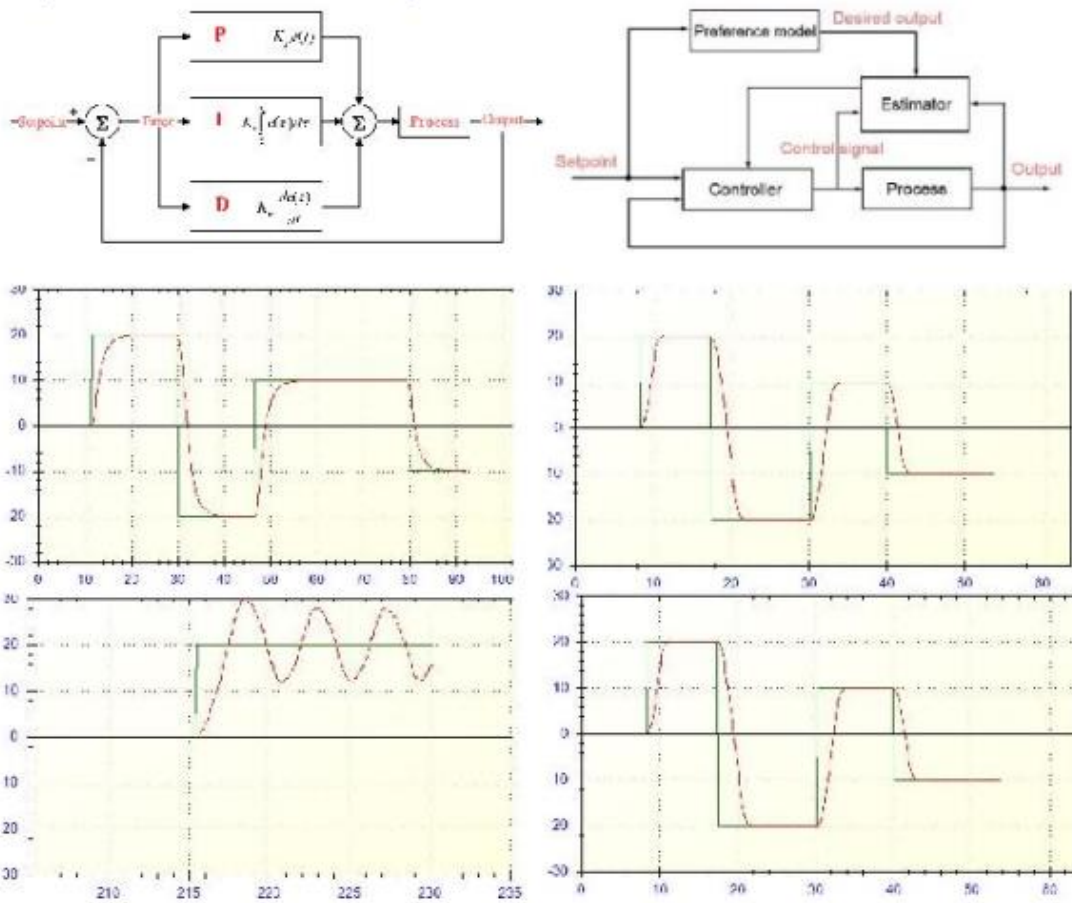
Động cơ DC được sử dụng rộng rãi trong công nghiệp vì tính chính xác, tin cậy, ổn định, giá thành hợp lý. Giải thuật điều khiển PID được sử dụng điều khiển động cơ DC vì tính đơn giản nhưng vẫn đạt được hiệu quả tương đối trong những trường hợp cố định.

Tuy nhiên, để điều khiển hiệu quả bộ điều khiển PID, chúng ta cần xác định đúng hệ số K_p , K_i , K_d , vì thế trong những hệ thống thay đổi liên tục, việc sử dụng bộ điều khiển PID thể hiện nhiều nhược điểm.

Trong những năm gần đây, bộ điều khiển tự chỉnh tỏ ra rất hứa hẹn trong những hệ thống mà thông số chưa biết hoặc luôn thay đổi bởi khả năng tự thích nghi, cho đáp ứng nhanh và chính xác song song với sự phát triển của những bộ vi điều khiển hiệu năng cao.

CƠ SỞ LÝ THUYẾT

1. GIỚI THIỆU CHUNG VỀ HAI BỘ ĐIỀU KHIỂN PID VÀ STR



Hình 1: So sánh giữa bộ điều khiển PID và STR

2. GIẢI THÍCH CHỨC NĂNG CÁC KHỐI TRONG GIẢI THUẬT STR

BỘ ƯỚC LƯỢNG (ESTIMATOR):

Tín hiệu vào: Tín hiệu điều khiển điện áp động cơ, tín hiệu ra thực tế, thông số động cơ từ mô hình chuẩn.

Tín hiệu ra: Thông số ước lượng của hệ thống.

Chức năng:

1. Ước lượng trạng thái của động cơ thông qua những thông số đọc về từ encoder và tín hiệu tham khảo từ mô hình chuẩn.
2. Tính toán giá trị sai số ước lượng trong lần điều khiển tiếp theo.

BỘ ĐIỀU KHIỂN (CONTROLLER):

Tín hiệu vào: Thông số ước lượng của hệ thống, tín hiệu đặt, tín hiệu ra thực tế.

Tín hiệu ra: Tín hiệu điều khiển điện áp động cơ.

Chức năng:

Tính toán tín hiệu điều khiển dựa trên kết quả của bộ ước lượng sao cho động cơ có đáp ứng hoàn hảo giống mô hình mẫu.

KHOI MÔ HÌNH CHUẨN (PREFERENCE MODEL):

Tín hiệu vào: Tín hiệu đặt.

Tín hiệu ra: Thông số động cơ từ mô hình chuẩn.

Chức năng:

1. Tính toán dựa trên những hàm của Matlab
2. Có đáp ứng hoàn hảo với vọt lố bằng không và thời gian xác lập nhanh nhất.

3. MÔ HÌNH TOÁN HỌC ĐIỀU KHIỂN VẬN TỐC ĐỘNG CƠ DC

Từ hàm truyền liên tục vị trí động cơ như trên, dựa vào lý thuyết rời rạc hóa, chúng ta thu được mô hình rời rạc vị trí động cơ như sau:

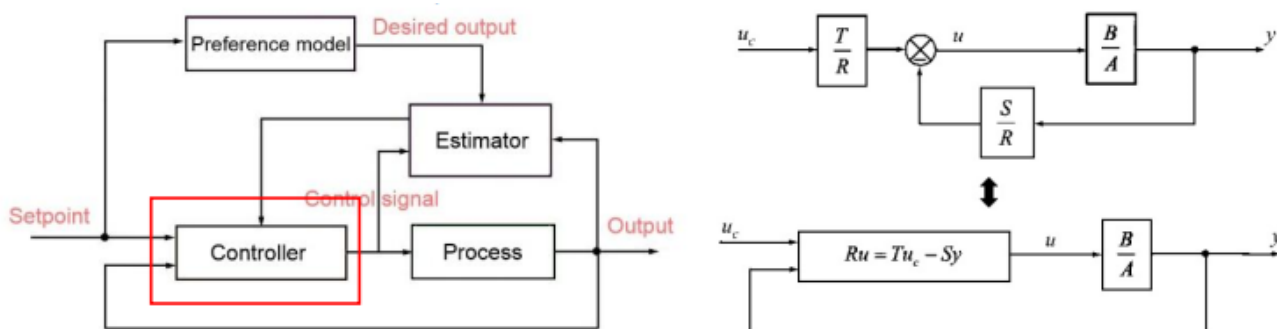
$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} u(z)$$

$$y_m(z) = \frac{B_m}{A_m} u_c(z) = \frac{b_{1m} z + b_{2m}}{z^2 + a_{1m} z + a_{2m}} u_c(z)$$

Bảng 1: Giải thích những thông số trong hàm truyền liên tục và rời rạc động cơ

Thông số	Giải thích
$u_c(t), u_c(k)$	Tín hiệu vận tốc mong muốn theo dạng liên tục và dạng rời rạc
$u(t), u(k)$	Tín hiệu điện áp ra từ bộ điều khiển theo dạng liên tục và dạng rời rạc
$y(t), y(k)$	Tín hiệu vận tốc thực tế theo dạng liên tục và dạng rời rạc
$y_m(t), y_m(k)$	Tín hiệu vận tốc từ mô hình mẫu theo dạng liên tục và dạng rời rạc
$a_1, a_2, a_3, b_1, b_2, b_3$	Thông số thực tế của mô hình toán học động cơ
$a_{1m}, a_{2m}, a_{3m}, b_{1m}, b_{2m}, b_{3m}$	Thông số mô hình mẫu của mô hình toán học động cơ

4. LÝ THUYẾT THIẾT KẾ BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN



Hình 2: Bộ điều khiển theo mô hình chuẩn

Lý thuyết:

1. Bài toán đặt ra là thiết kế R, T, S để đáp ứng của hệ kín $\frac{B}{A}$ bám hoàn hảo theo mô hình chuẩn $\frac{B_m}{A_m}$.

2. So sánh hàm truyền hệ kín với mô hình chuẩn:

$$y(t) = \frac{BT}{AR + BS} u_c(t)$$

$$y_m(t) = \frac{B_m}{A_m} u_c(t)$$

3. Để đạt được đáp ứng vòng kín mong muốn, cần có điều kiện:

- Phương trình đặc trưng của hệ kín phải có các cực trùng với các cực của mô hình chuẩn tức là $AR+BS$ phải chia hết cho các cực của mô hình chuẩn, tức là $AR+BS$ phải chia hết cho A_m .
- Các zero nằm bên trái mặt phẳng phức của B phải được triệt tiêu bởi các cực của hệ kín. Giả sử có thể phân tích $B = B^+ B^-$ (với B^+ gồm các zero bên trái mặt phẳng phức), chúng ta cần điều kiện $AR + BS$ chia hết cho B^+ .

4. Từ đó ta có phương trình Diophantine:

$$AR + BS = A_0 A_N B^+$$

5. Để có thể khử B^+ , R phải có dạng:

$$R = R_1 B^+$$

$$\Rightarrow AR_1 B^+ + BS = A_0 A_N B^+$$

$$\Rightarrow AR_1 + BS = A_0 A_N$$

6. Với các điều kiện trên hàm truyền hệ kín trở thành:

$$y(t) = \frac{B^+ B^- T}{A_0 A_m B^+} u_c(t)$$

7. Rút gọn và so sánh với mô hình mẫu, ta có:

$$y(t) = \frac{B^- T}{A_0 A_m} u_c(t)$$

$$y_m(t) = \frac{B_m}{A_m} u_c(t)$$

8. Điều kiện cần để hàm truyền hệ kín đúng bằng mô hình mẫu là:

$$B_m = B^- B'_m$$

$$T = A_0 B'_m$$

9. Điều kiện đảm bảo tồn tại lời giải bài toán điều khiển theo mô hình chuẩn là:

$$\text{Bậc } A_0 \geq 2 \text{ Bậc } A - \text{Bậc } A_m - \text{Bậc } B^+ - 1$$

$$\text{Bậc } A_m - \text{Bậc } B_m \geq \text{Bậc } A - \text{Bậc } B$$

10. Phương pháp đơn giản tìm nghiệm phương trình Diophantine:

- Chọn bậc của đa thức R và S phù hợp.
- Cân bằng các hệ số của phương trình Diophantine sẽ tìm được các hệ số của R và S.

BỘ VI XỬ LÝ

Chúng ta nên dùng những dòng chip ARM®Cortex™M vì những dòng chip này có xung nhịp tương đối từ 80MHz đến 500Mhz, cùng bộ tính toán số thực có sẵn trong chip giúp tính toán nhanh giải thuật ước lượng nhân chia nhiều ma trận số thực trong ít nhất là 5ms, chu kì trung bình để có thể điều khiển mượt động cơ DC mà những dòng chip bình thường như 8051, PIC16, Arduino không thể làm được điều đó. Mặt khác, chip ARM hỗ trợ thư viện rất mạnh mẽ với giá cả cạnh tranh khiến chúng ta đơn giản hóa phần nào quá trình thực hiện bộ điều khiển tự chỉnh.

1. Những module cần có của vi điều khiển để ứng dụng bộ điều khiển thích nghi gồm có: Module System tick hoặc timer để tạo ra chính xác chu kì điều khiển động cơ là 5ms hoặc 10ms.
2. Ngoài ra cần phải có 1 bộ PWM để điều khiển điện áp động cơ.
3. Một bộ đọc Encoder Quadrature bằng phần cứng.
4. Một chân GPIO để điều khiển hướng động cơ.
5. Một module UART để truyền tín hiệu hai chiều với máy tính hoặc với module khác.

1. VI ĐIỀU KHIỂN STM32F407 TRÊN KIT STM32F4 DISCOVERY

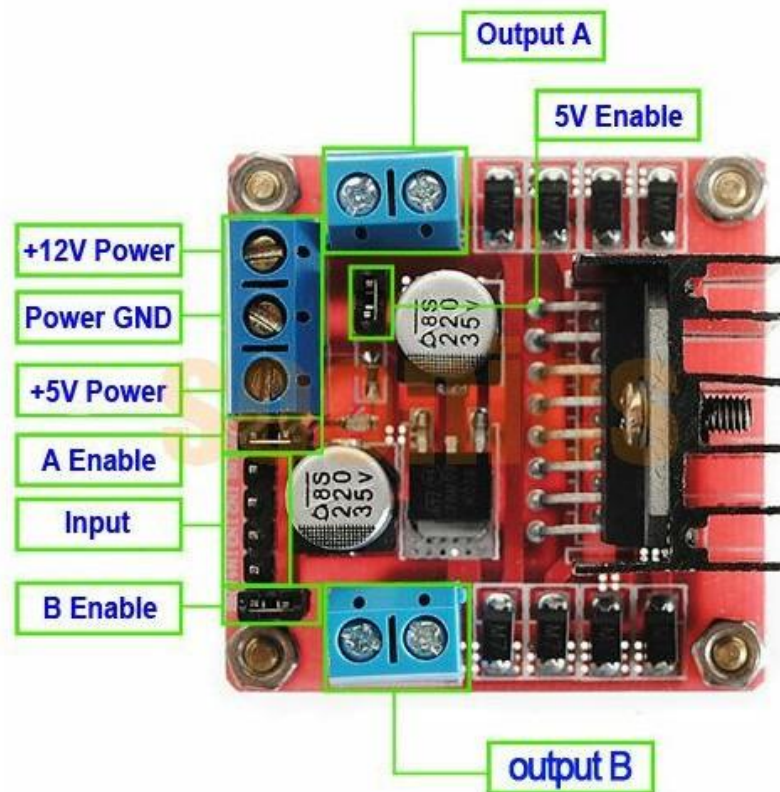


Hình 3: Vi điều khiển STM32F407VG

1. Vi điều khiển STM32F407VG với tính năng 32-bit ARM Cortex-M4F lõi, 1 MB Flash, 192 KB bộ nhớ RAM trong một gói LQFP100
2. On-board ST-LINK/V2 với chế độ lựa chọn chuyển sang sử dụng các bộ như một ST- LINK/V2 độc lập (với kết nối SWD cho lập trình và gỡ lỗi)
3. Điện nguồn cung cấp: 3 V và 5 V
4. LIS302DL, cảm biến chuyển động ST MEMS, 3 trục đầu ra kỹ thuật số gia tốc kế
5. MP45DT02, ST MEMS cảm biến âm thanh, microphone kỹ thuật số omni-directional
6. CS43L22, DAC âm thanh tích hợp trình điều khiển lớp D loa
7. LD1 (màu đỏ / xanh lá cây) để giao tiếp USB, LD2 (màu đỏ) cho 3,3 V điện trên
8. Bốn đèn LED sử dụng, LD3 (màu cam), LD4 (màu xanh), LD5 (màu đỏ) và LD6 (màu xanh)

9. 2 USB OTG đèn LED LD7 (màu xanh) và VBus LD8 (màu đỏ) quá dòng
10. Hai nút bấm (người sử dụng và reset)
11. USB OTG FS với cổng nối micro-AB
12. Header mở rộng cho tất cả các LQFP100 I / O để kết nối nhanh chóng và dễ dàng tạo mẫu bảng thăm dò
13. DSP ứng dụng xử lý tính toán số học.

2. MẠCH CẦU H



Hình 4: L298N

Thông số kỹ thuật:

- Driver: L298N tích hợp hai mạch cầu H.
- Điện áp điều khiển: +5 V ~ +12 V
- Dòng tối đa cho mỗi cầu H là: 2A (=>2A cho mỗi motor)
- Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
- Dòng của tín hiệu điều khiển: 0 ~ 36mA (Arduino có thể chơi đến 40mA nên khỏe re nhé các bạn)
- Công suất hao phí: 20W (khi nhiệt độ T = 75 °C)
- Nhiệt độ bảo quản: -25 °C ~ +130 °C

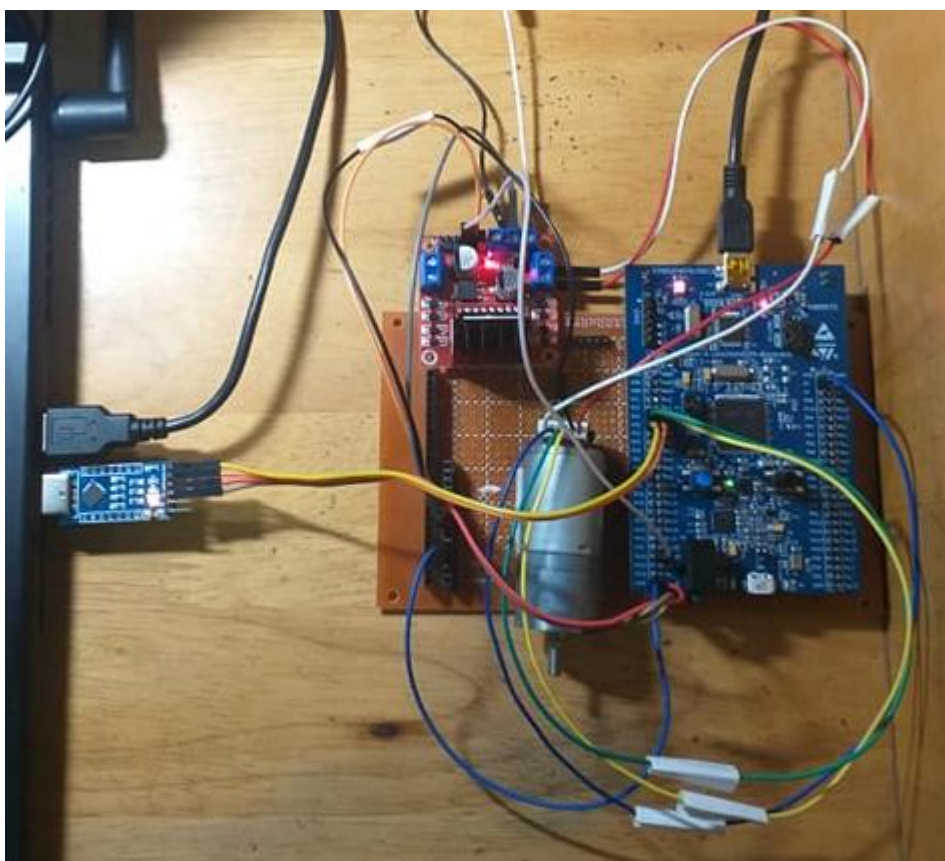
L298 gồm các chân:

- 12V power, 5V power. Đây là 2 chân cấp nguồn trực tiếp đến động cơ.
 - Bạn có thể cấp nguồn 9-12V ở 12V.
 - Bên cạnh đó có jumper 5V, nếu để như hình ở trên thì sẽ có nguồn 5V ra ở cổng 5V power, ngược lại thì không. Nếu để như hình thì ta chỉ cần cấp nguồn 12V vô ở 12V power là có 5V ở 5V power, từ đó cấp cho Arduino
- Power GND chân này là GND của nguồn cấp cho động cơ.
- Gồm có 4 chân Input. IN1, IN2, IN3, IN4.
- Output A: nối với động cơ A. chú ý chân +, -.

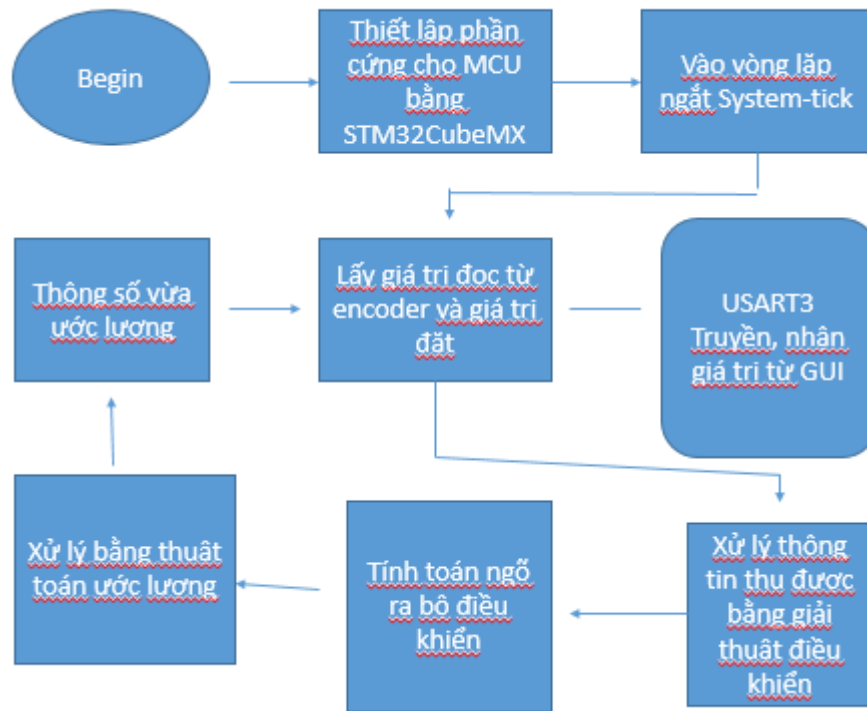
Cầu H điều khiển động cơ với thời gian delay ngắn hơn từ delay micro giây xuống còn nano giây, giúp cho giải thuật điều khiển gần hơn với lý thuyết mô phỏng, chính xác và hiệu quả hơn.

3. SẢN PHẨM CỦA NHÓM ĐÃ THỰC HIỆN

Link kiện	Số lượng
STM32F4	1
Động cơ DC5V	1
Mạch cầu H L298N	1
Cổng kết nối USB STM32 đến laptop	1
Cổng nối tiếp UART	1
Dây nối	

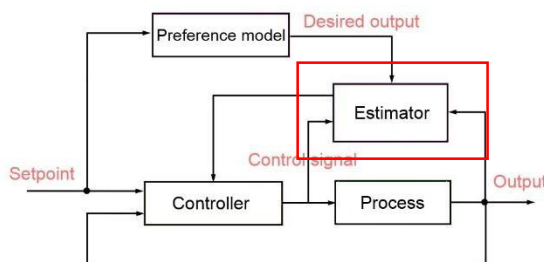


XÂY DỰNG GIẢI THUẬT



Hình 5: Sơ đồ khối quá trình lập trình nhúng trong vi điều khiển

1. BỘ ƯỚC LƯỢNG BÌNH PHƯƠNG TỐI THIỂU ĐỆ QUY



Hình 6: Bộ ước lượng bình phương tối thiểu đệ quy

1.1 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

Ý nghĩa của giải thuật này là xác định được vận tốc ước lượng của động cơ là $\hat{y}(k)$ thông qua các thông số thu về từ encoder và setpoint trong ma trận $\Phi^T(k)$ và thông số ước lượng mô hình rời rạc trong ma trận $\hat{\theta}(k)$. Thông qua sự chênh lệch giữa vận tốc ước lượng và vận tốc thực tế, luật điều khiển sẽ đưa ra output phù hợp.

$$\hat{y}(k) = \varphi^T(k) \hat{\theta}(k)$$

$$\begin{cases} y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) \\ \theta(k) = [a_1(k) \ a_2(k) \ b_1(k) \ b_2(k)]^T \\ \varphi(k) = [-y(k-1) \ -y(k-2) \ u(k-1) \ u(k-2)]^T \end{cases}$$

Bước 1: Khởi tạo ma trận thông số khởi đầu $\theta(0)$ và ma trận covariance $P(0)$

$$\theta(0) = \text{random}(4,1)$$

$$P(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bước 2: Cập nhật ma trận trạng thái hệ thống $\varphi(k)$

$$\varphi(k) = [-y(k-1) \ -y(k-2) \ u(k-1) \ u(k-2)]^T$$

Bước 3: Tính toán ma trận gain Kalman $L(k)$ mới theo công thức:

$$L(k) = \frac{P(k-1)\varphi(k)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)}$$

Bước 4: Tính toán ma trận sai số ước lượng $\epsilon(k)$:

$$\epsilon(k) = y(k) - \varphi^T(k) \hat{\theta}(k-1)$$

Bước 5: Tính toán ma trận trạng thái hệ thống $\varphi(k)$ mới và lấy các hệ số ước lượng trạng thái động cơ

$$\hat{\theta}(k) = \hat{\theta}(k-1) + L(k)\epsilon(k)$$

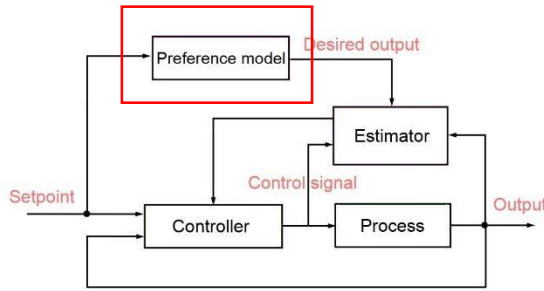
$$\hat{\theta}(k) = [\hat{a}_1(k) \ \hat{a}_2(k) \ \hat{b}_1(k) \ \hat{b}_2(k)]^T$$

Bước 6: Tính toán ma trận covariance P mới theo công thức:

$$P(k) = \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)\varphi^T(k)\varphi(k)P(k-1)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \right] = \frac{1}{\lambda} [P(k-1) - L(k)\varphi(k)P(k-1)]$$

Bước 7: Quay về Bước 2, bước qua lần lấy mẫu và điều khiển mới

2. KHỐI MÔ HÌNH CHUẨN



Hình 7: Khối mô hình chuẩn

3. GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

$$\frac{Y_m(s)}{U_c(s)} = G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

$$\frac{Y_m(z)}{U_c(z)} = G_d(z) = \frac{b_{1m}z + b_{2m}}{z^2 + a_{1m}z + a_{2m}}$$

Bước 1: Thiết kế mô hình chuẩn:

- Độ vọt lố (Percentage of Overshoot) là %:

$$\begin{cases} POT = e^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}} \\ POT = 0\% \end{cases}$$

$$\Rightarrow \xi = 1$$

- Thời gian xác lập T_{qd} là 1 giây:

$$\begin{cases} T_{qd} = \frac{4}{\omega_n \xi} \\ T_{qd} = 1 \end{cases}$$

$$\Rightarrow \omega_n = 4$$

Bước 2: Tính toán các hệ số mô hình chuẩn a_{1m} , a_{2m} , b_{1m} , b_{2m} bằng hàm Matlab:

```
>> G = tf([4*4], conv([1 2*4 4*4], [1]))
```

$$\frac{0.000779 z + 0.0007585}{z^2 - 1.922 z + 0.9231}$$

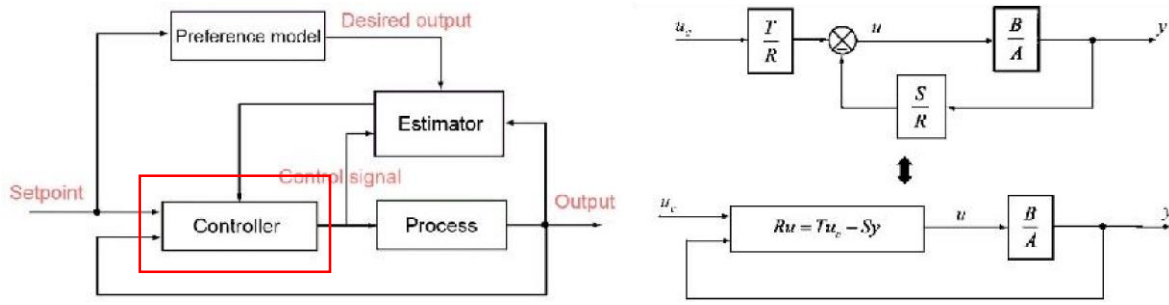
Sample time: 0.01 seconds

Discrete-time transfer function.

Ta có hàm truyền rời rạc của mô hình chuẩn:

$$\frac{Y_m(z)}{U_c(z)} = G_d(z) = \frac{b_{1m}z + b_{2m}}{z^2 + a_{1m}z + a_{2m}} = \frac{0.000779z + 0.0007585}{z^2 - 1.922z + 0.9231}$$

4. BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN



Hình 9: Bộ điều khiển theo mô hình chuẩn

5. GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

Đối tượng:

$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2}$$

Luật điều khiển

$$u = \left(\frac{b_{1m}}{b_1} + \frac{b_{2m}}{b_1} z^{-1} \right) u_c - \left[\left(\frac{a_{1m}}{b_1} - \frac{a_1}{b_1} \right) + \left(\frac{a_{2m}}{b_1} - \frac{a_2}{b_1} \right) z^{-1} \right] y - \left(\frac{b_2}{b_1} z^{-1} \right) u$$

6. HỆ SỐ QUÊN THAY ĐỔI THEO THÔNG SỐ CỦA HỆ THỐNG

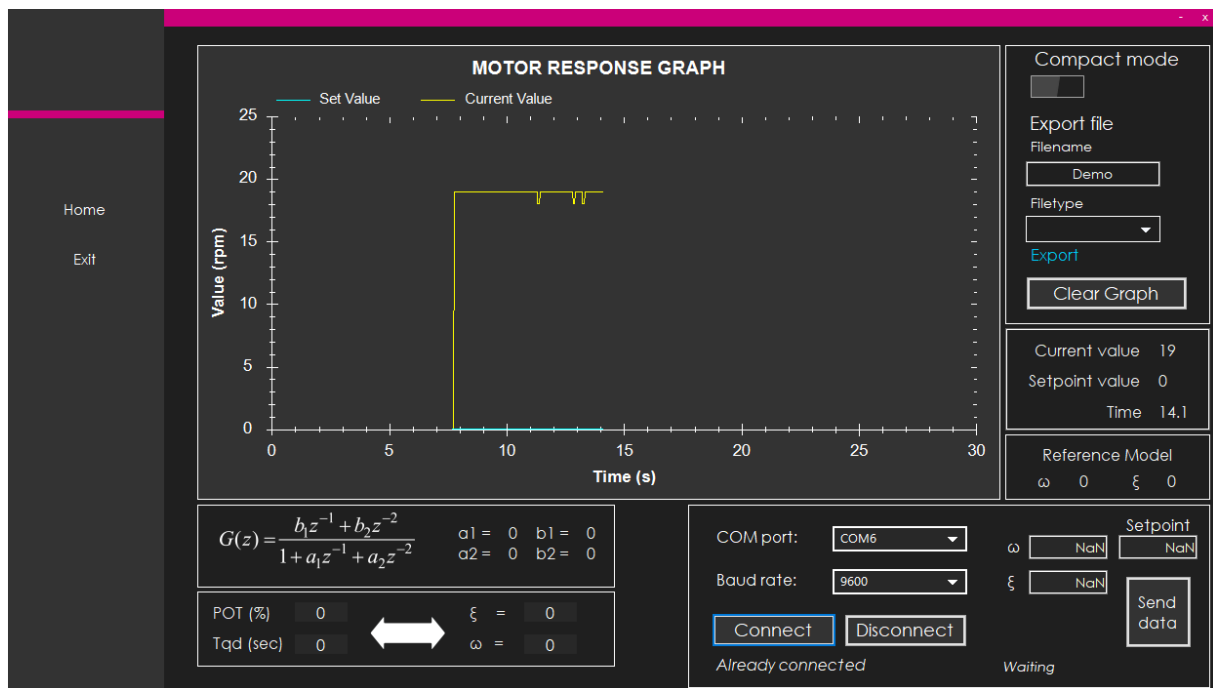
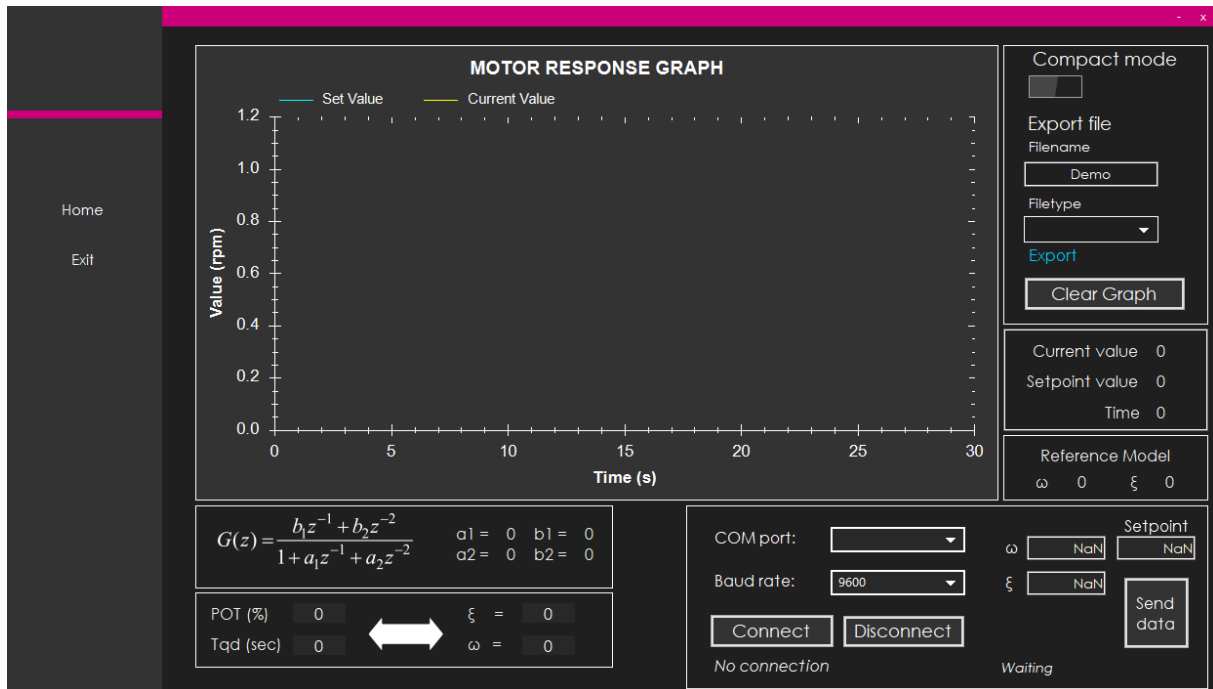
Hệ số quên trong công thức bộ ước lượng cơ bản có hệ số λ là một hằng số. Trước hết, chúng ta cần làm rõ chức năng vô cùng quan trọng của hệ số quên trong giai đoạn điều khiển này:

Hệ số quên làm cho những thông số của ma trận covariance P không trở nên quá nhỏ làm mất tính thích nghi. Sự đặc trưng của tính thích nghi thể hiện qua thông số N gọi là độ dài của bộ nhớ thích nghi: $N = \frac{1}{1-\lambda}$

Hệ thống giải thuật thích nghi dùng những thông số từ N mẫu trước đó để tính toán và ước lượng thông số hiện tại của động cơ. Bởi vì đây là giải thuật hồi quy, nó không thể thích nghi và tính toán tất cả những thông số của động cơ từ lúc hoạt động đến vô cùng, đơn giản vì không một vi xử lý nào có thể tính toán số lượng phép tính tăng tuyến tính đến vô cực cả.

Ở bảng thông tin thông số trước, giải thuật cơ bản khuyến nghị hệ số λ là hằng số từ 0.95 đến 0.998 tùy hệ thống tức là giải thuật sẽ thích nghi từ 20 đến 500 thông số trước đó của động cơ.

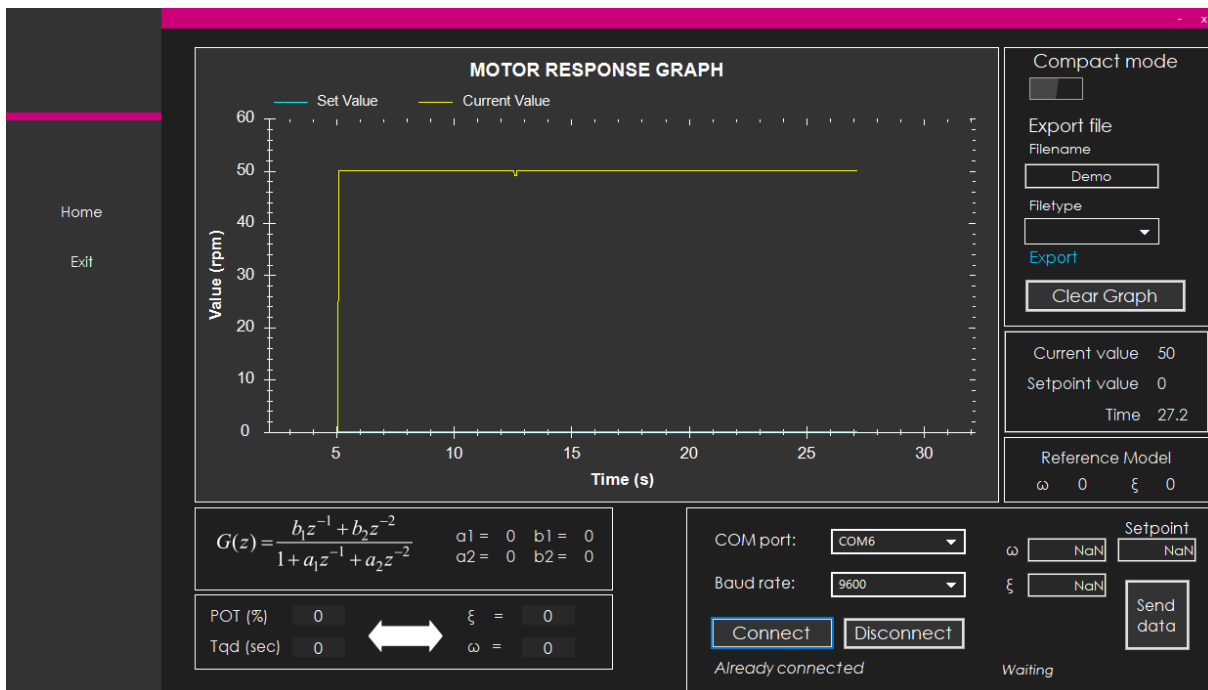
THIẾT KẾ GIAO DIỆN ĐIỀU KHIỂN DÙNG C# FORM



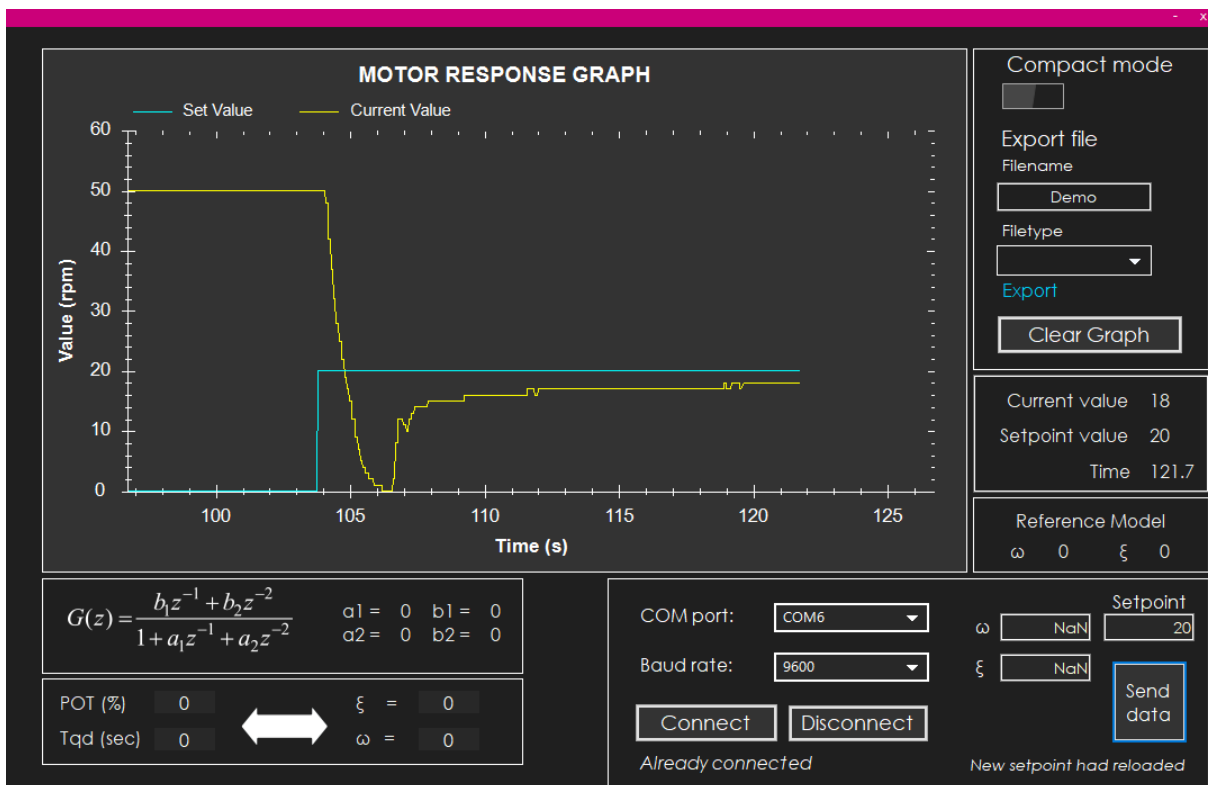
KẾT QUẢ VÀ NHẬN XÉT

CHẤT LƯỢNG ĐIỀU KHIỂN ĐỘNG CƠ

- Quan sát khi tốc độ đặt mặc định đặt ở phần code vi điều khiển là 50 rpm

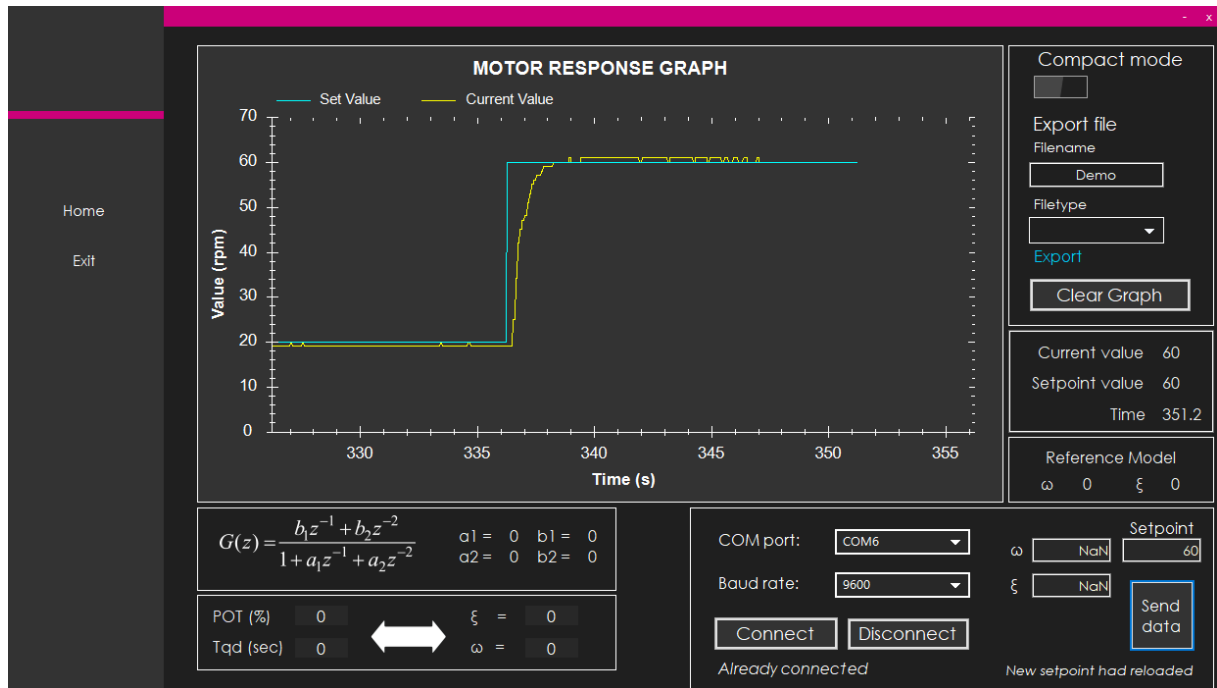


- Giảm tốc độ đặt xuống còn 20 rpm



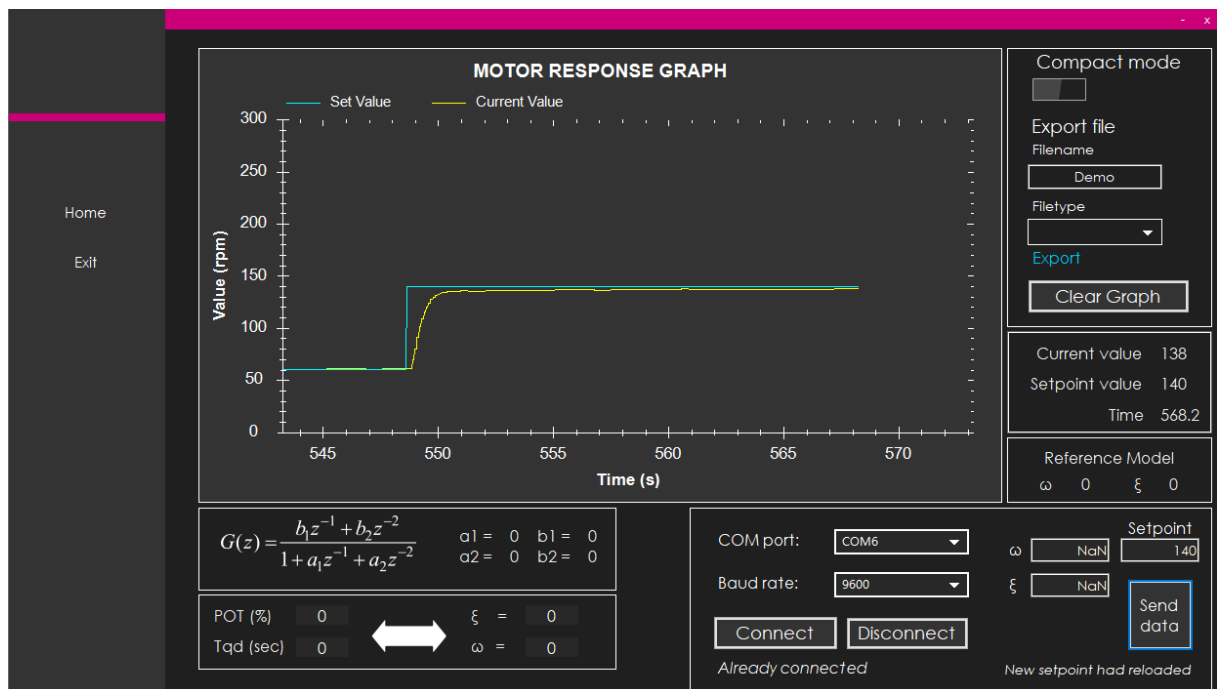
Nhận xét: Đáp ứng không có vọt lố và sai số xác lập là -2 rpm.

- Tăng tốc độ đặt lên 60 rpm



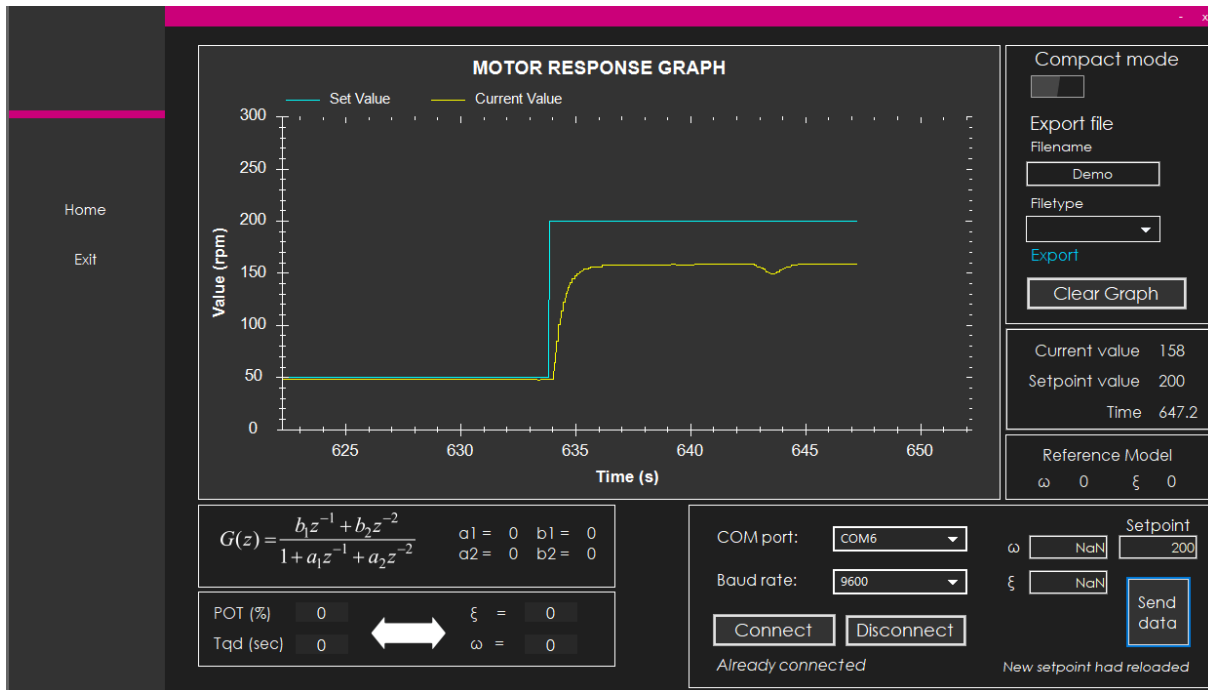
Nhận xét: Đáp ứng không có vọt lố và sai số xác lập là 0 rpm.

- Tăng tốc độ đặt lên 140 rpm.



Nhận xét: Đáp ứng không có vọt lố và sai số xác lập là -2 rpm.

- Tốc độ giới hạn động cơ của nhóm là khoảng 158 rpm:



NHẬN XÉT CHẤT LƯỢNG HỆ THỐNG

- Hệ thống đạt được chất lượng mô hình chuẩn đã thiết kế, tất cả các trường hợp kiểm tra hệ thống đã thiết kế đều cho kết quả không có vọt lố.
- Sai số xác lập rất ít, lớn nhất là ± 2 rpm.

HẠN CHẾ VÀ HƯỚNG CẢI TIẾN

- Hiện tại sản phẩm chỉ hạn chế ở điều khiển tốc độ của động cơ. Trong tương lai, nhóm sẽ hoàn thiện và điều khiển được vị trí – có hàm truyền bậc cao hơn, phải giải quyết phức tạp hơn.
- Hiện tại nhóm chỉ có truyền thông dữ liệu qua từng byte, chưa có chuẩn truyền thông theo khung dữ liệu, hiện tại chỉ có thể truyền/ nhận một thông số (cụ thể ở đây là tốc độ). Trong tương lai nhóm sẽ thiết kế tiêu chuẩn truyền nhận data đóng khung dữ liệu để có thể điều khiển và hiển thị nhiều thông số.
- Chính chu mạch có thể trở thành một sản phẩm đẹp mắt, bố trí dây nối và kết cấu cơ khí chắc chắn hơn.